BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS
- BLANK PAGES

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

(12) UK Patent Application (19) GB (11) 2 338 574 (13) A

(43) Date of A Publication 22.12.1999

(21) Application No 9825088.9

(22) Date of Filing 17.11.1998

(30) Priority Data
(31) 98023201 (32) 19.06.1998 (33) KR

(51) INT CL[6]
G06F 9/445

(52) UK CL (Edition Q)
G4A AFL

(56) Documents Cited
US 5371871 A US 5212633 A

(58) Field of Search
UK CL (Edition Q) G4A AFL
INT CL[6] G06F 9/445

(71) Applicant(s)
Samsung Electronics Co Limited
(Incorporated in the Republic of Korea)
416 Maetan-dong, Paldal-gu, Suwon-city, Kyungki-Do,
Republic of Korea

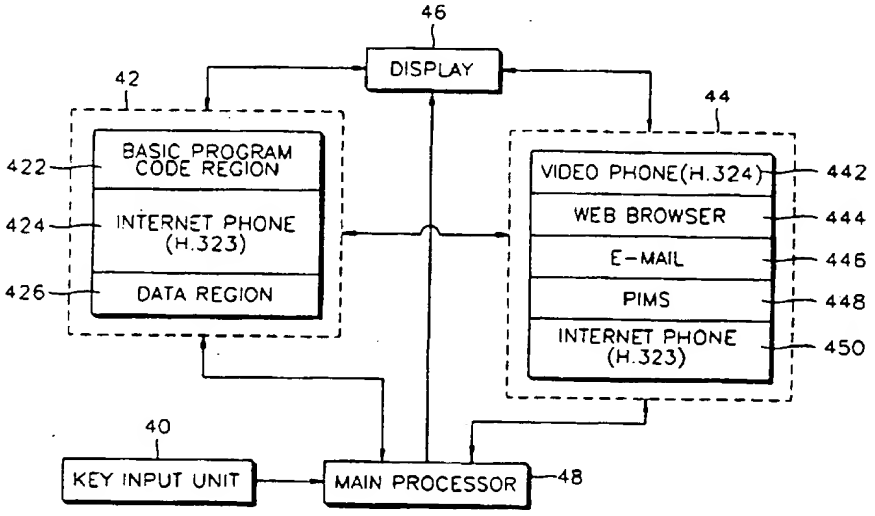
(72) Inventor(s)
Jae-Sik Hwang

(74) Agent and/or Address for Service
Appleyard Lees
15 Clare Road, HALIFAX, West Yorkshire, HX1 2HY,
United Kingdom

(54) Abstract Title
Web video telephone with application program swapping

(57) The size of a DRAM 42 in a video telephone is reduced by storing application programs and associated data in an auxiliary flash memory 44. An application program and data are loaded into respective portions 424 and 426 of main memory 42 when the application is selected by a user from a menu on display 46. The selection is entered through key input unit 40. Applications may include a video telephone program using the H.324 protocol 442, an internet telephone using the H.323 protocol 450, a web browser 444, an e-mail facility 446, or a personal information management system 448.

FIG. 4



GB 2 338 574 A

FIG. 1A (PRIOR ART)

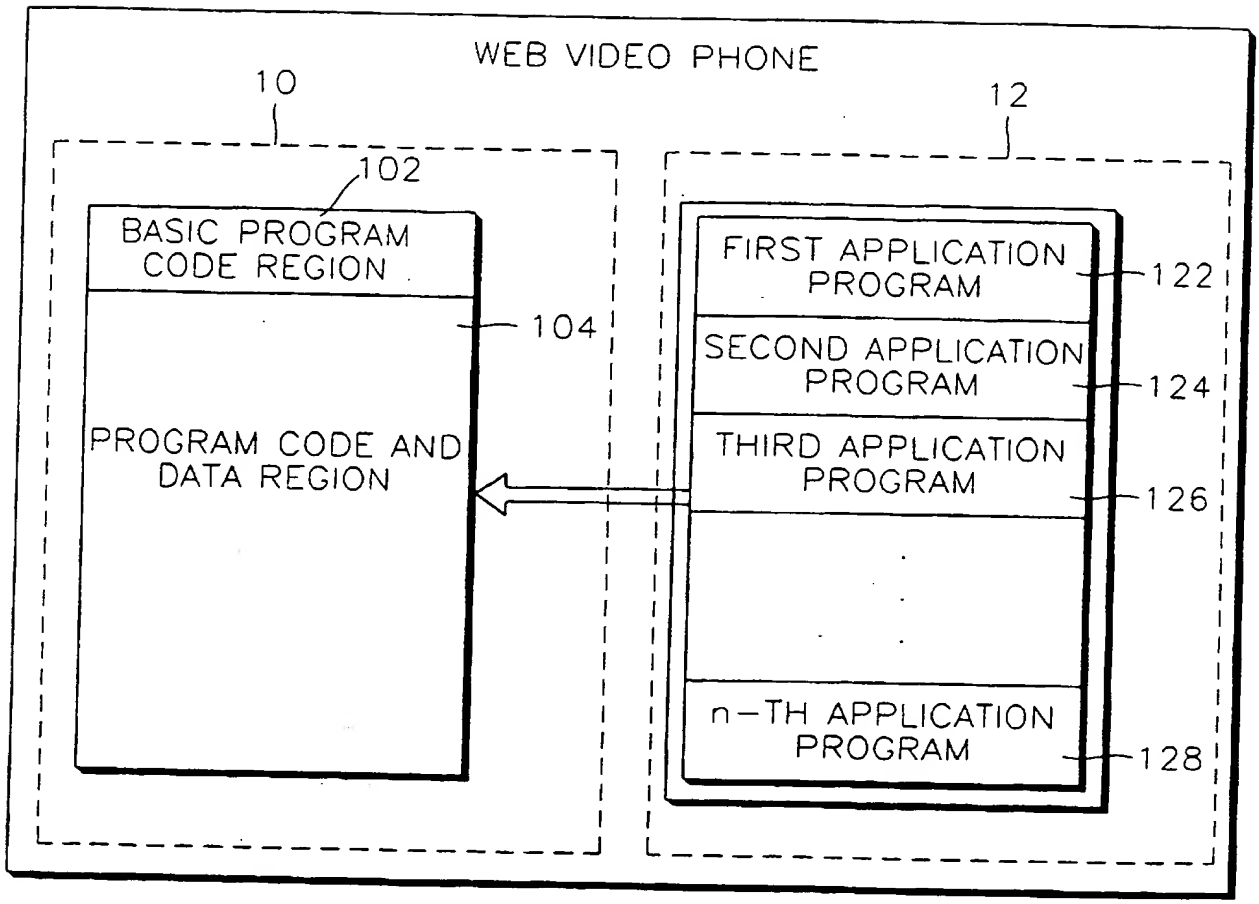


FIG. 2

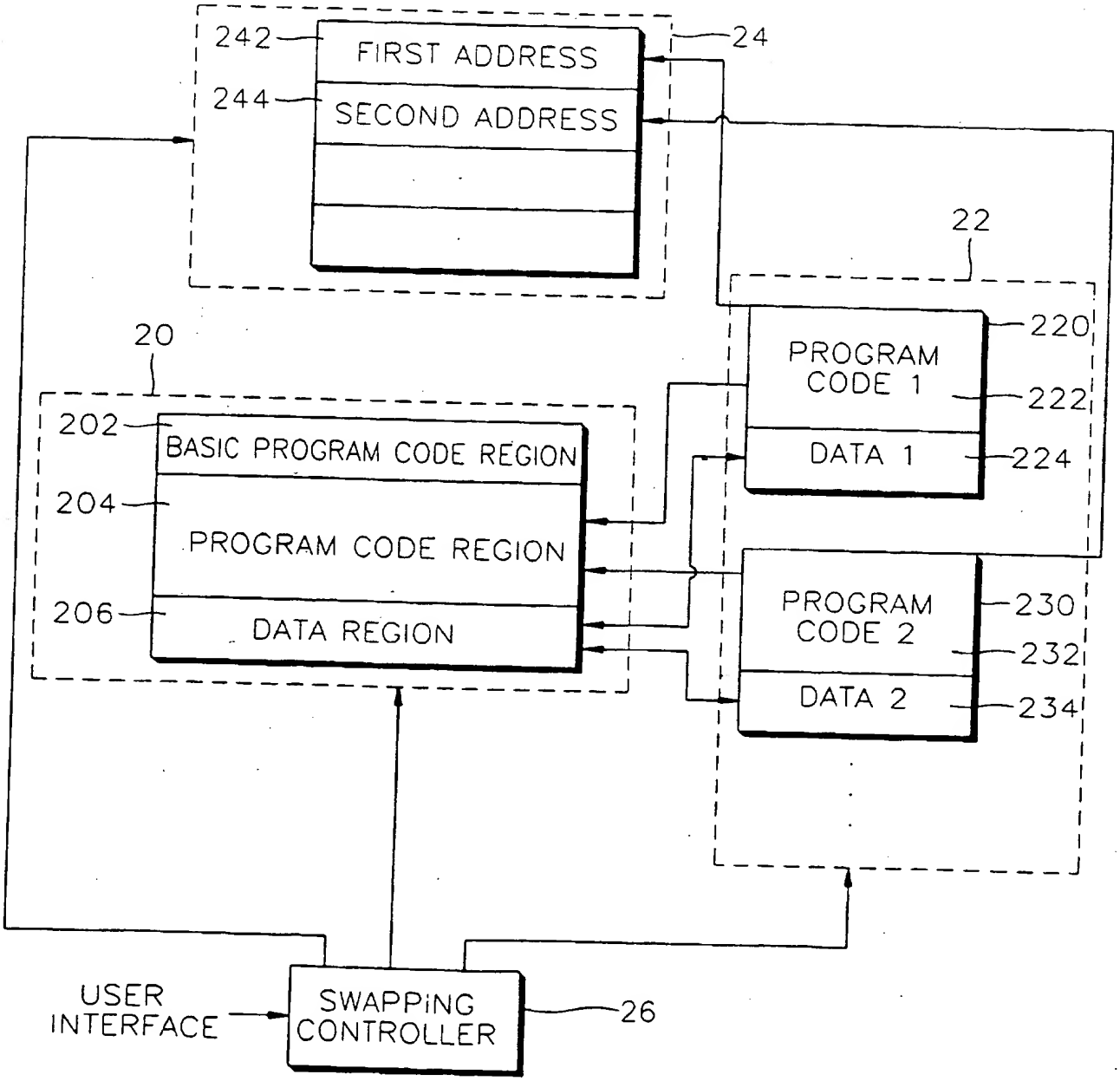

242
FIRST ADDRESS
244
SECOND ADDRESS
24
22
20
PROGRAM CODE 1
220
222
DATA 1
224
202
BASIC PROGRAM CODE REGION
204
PROGRAM CODE REGION
206
DATA REGION
PROGRAM CODE 2
230
232
DATA 2
234
USER INTERFACE
SWAPPING CONTROLLER
26

FIG. 3

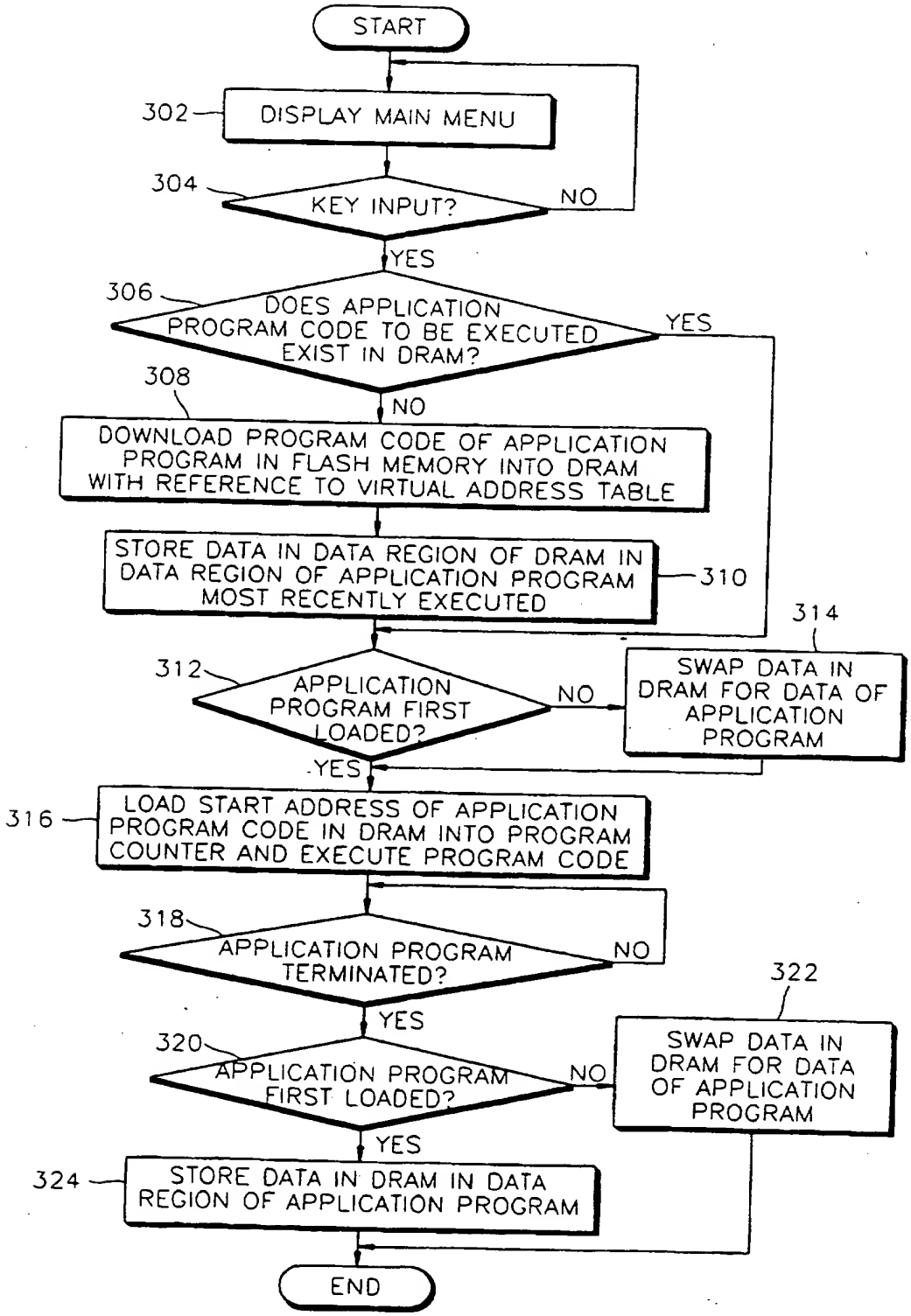

START
302
DISPLAY MAIN MENU
304
KEY INPUT?
NO
YES
306
DOES APPLICATION PROGRAM CODE TO BE EXECUTED EXIST IN DRAM?
YES
308
NO
DOWNLOAD PROGRAM CODE OF APPLICATION PROGRAM IN FLASH MEMORY INTO DRAM WITH REFERENCE TO VIRTUAL ADDRESS TABLE
STORE DATA IN DATA REGION OF DRAM IN DATA REGION OF APPLICATION PROGRAM MOST RECENTLY EXECUTED
310
314
312
APPLICATION PROGRAM FIRST LOADED?
NO
SWAP DATA IN DRAM FOR DATA OF APPLICATION PROGRAM
YES
316
LOAD START ADDRESS OF APPLICATION PROGRAM CODE IN DRAM INTO PROGRAM COUNTER AND EXECUTE PROGRAM CODE
318
APPLICATION PROGRAM TERMINATED?
NO
YES
322
320
APPLICATION PROGRAM FIRST LOADED?
NO
SWAP DATA IN DRAM FOR DATA OF APPLICATION PROGRAM
YES
324
STORE DATA IN DRAM IN DATA REGION OF APPLICATION PROGRAM
END

FIG. 4

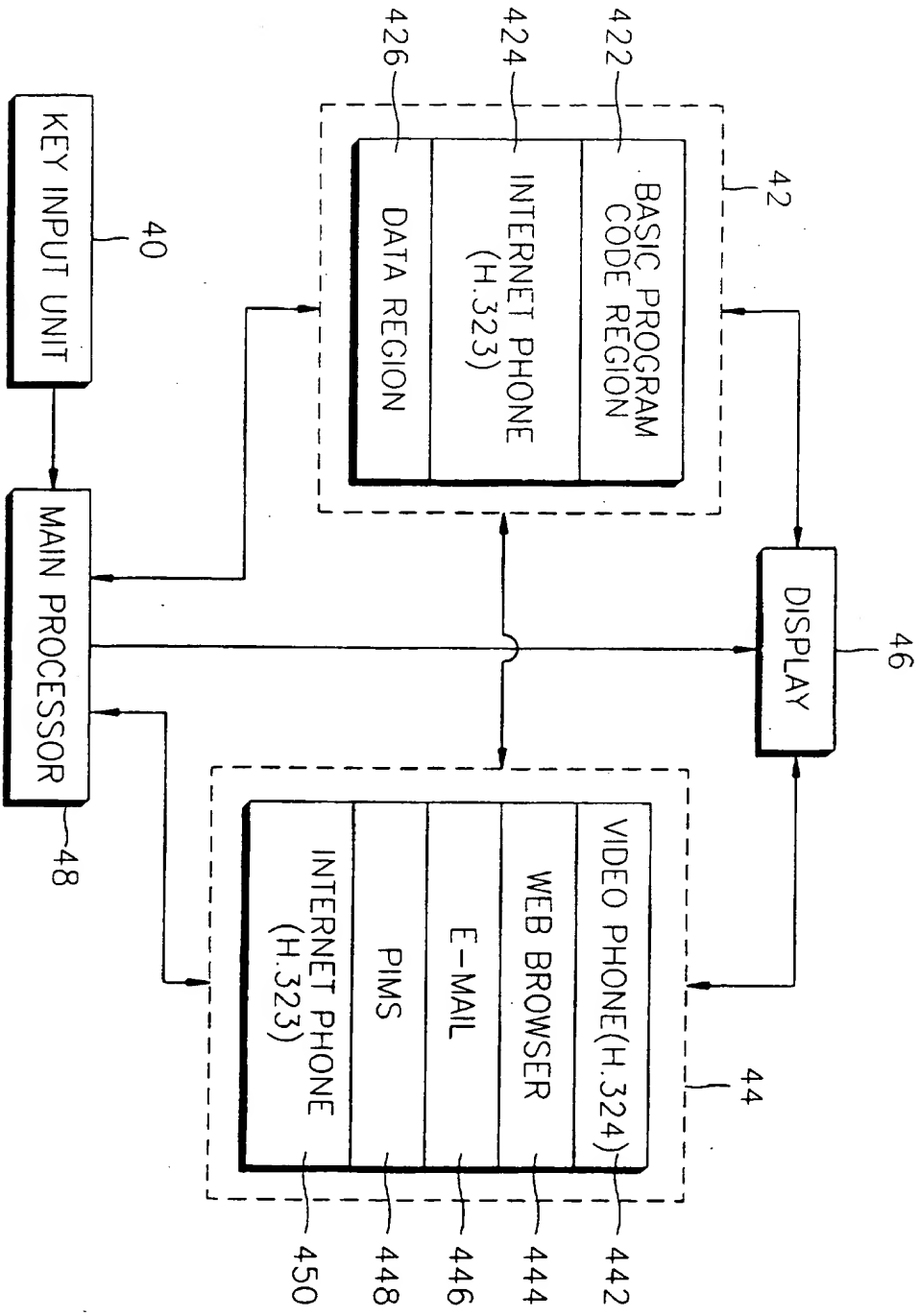
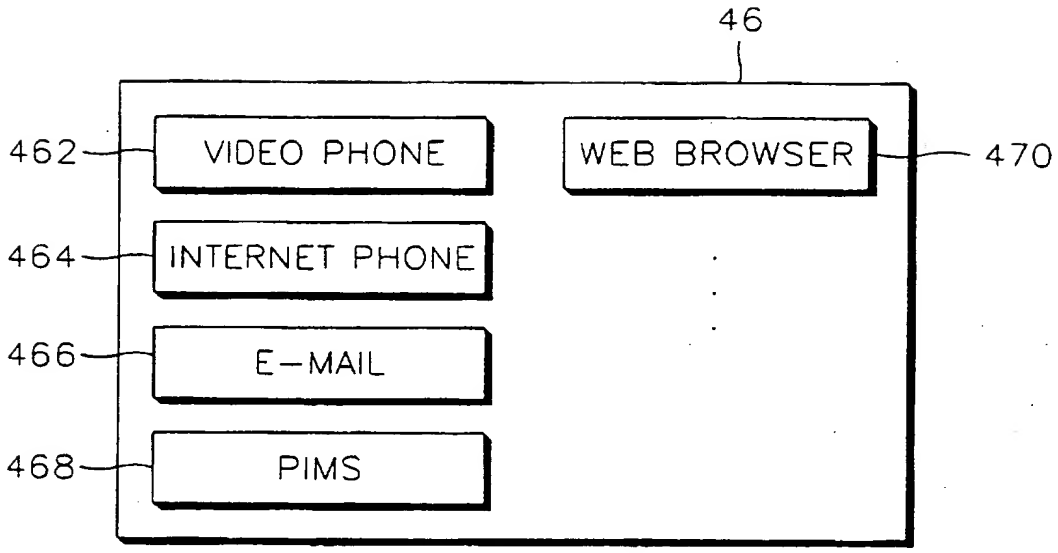

KEY INPUT UNIT
40
MAIN PROCESSOR
48
42
BASIC PROGRAM CODE REGION
INTERNET PHONE (H.323)
DATA REGION
422
424
426
DISPLAY
46
44
VIDEO PHONE(H.324)
WEB BROWSER
E-MAIL
PIMS
INTERNET PHONE (H.323)
442
444
446
448
450

FIG. 5


46
462
VIDEO PHONE
WEB BROWSER
470
464
INTERNET PHONE
466
E-MAIL
468
PIMS

APPARATUS AND METHOD FOR SWAPPING BETWEEN MAIN MEMORY AND AUXILIARY MEMORY, AND WEB VIDEO PHONE ADOPTING THE APPARATUS

The present invention relates to a swapping apparatus and method, and more particularly, to an apparatus and method for swapping between a main memory and an auxiliary memory, and a web video phone adopting the apparatus.

Swapping is the exchange of content stored in a main memory with content stored in an auxiliary memory. Here, swapping includes swap-in and swap-out. Swap-in is the transfer of a program having high priority from the auxiliary memory to the main memory. Swap-out is the transfer of a program having low priority from the main memory to the auxiliary memory.

Figure 1 is a block diagram of a conventional web video phone which does not adopt a swapping apparatus.

Referring to Figure 1, a DRAM 10, being a type of main memory, has a basic program code region 102 for storing a real time OS (operating system) code and a main menu, and a program code and data region 104 for swapping application programs in a flash memory 12.

The flash memory 12, being a type of auxiliary memory, stores application programs (122,124,126,... and 128) static-linked to an OS library. In the web video phone of Figure 1, all application programs stored in the flash memory 12 are downloaded into the DRAM 10 all at once in the power starting sequence.

However, the web video phone of Figure 1 has the following problems.

First, since the application programs static-linked to the OS library cannot be separated, they are downloaded into the DRAM 10 all at once, which requires a DRAM having a large capacity. That is, the DRAM 10 must have a capacity of at least 4M byte to download four application programs, each of 1M byte size from the flash memory 12 to the DRAM 10. Second, in the static linking of application programs stored in the flash memory, even when only one application program is changed, the other application programs must be recompiled and linked, which results in difficult management of the application programs.

It is an aim of at least preferred embodiments of the present invention to provide an improved apparatus and method of managing data transfers between main and auxiliary memories, and to a web video phone adopting the apparatus.

According to a first aspect of the present invention, there is provided an apparatus for swapping between a main memory and an auxiliary memory, comprising: an auxiliary memory for storing a plurality of application programs including program codes and data; a virtual address table for storing a start address of each application program stored in the auxiliary memory; a main memory having a program code region and a data region for storing a program code downloaded from the auxiliary memory and data swapped for the auxiliary memory; and a swapping controller for controlling the program codes of the programs to be downloaded into the program code region of the main memory with reference to addresses stored in the virtual address table, and controlling the swapping between the data of the programs and data stored in the data region of the main memory, in response to a request by a user interface to execute the application programs.

According to a second aspect of the present invention, there is provided a method of swapping between a main memory and an auxiliary memory, comprising the steps of: (a) determining whether the program code of an application program to be executed exists in a data region of the main memory; (b) downloading a program code stored in the auxiliary memory into a program code region of the main memory, when it is determined in step (a) that the program code does not exist; (c) swapping data of an application program stored in the auxiliary memory for data stored in the data region of the main memory; and (d) loading to a program counter a start address of the program downloaded into the main memory in step (b) and executing the program code.

According to a third aspect of the present invention, there is provided a web video phone wherein a DRAM and a flash memory are swapped, comprising: a key input unit for inputting a command of execution of an application program; a flash memory for storing application programs such as at least a video phone using the H.324 protocol, an Internet phone using the H.323 protocol, a web browser, an e-mail, and a personal information management system (PIMS); a DRAM including a program code region and a data region for storing program codes downloaded in the flash memory and data swapped for the flash memory; and a main processor for controlling the program code of each application program stored in the flash memory to be stored in the program region of the DRAM and controlling the swapping between the data of each application program stored in the flash memory and the data stored in the data region of the DRAM, in response to an execution command of an application program input through the key input unit.

For a better understanding of the invention, and to show how embodiments of the same may be carried into effect, reference will now be made, by way of example, to the accompanying diagrammatic drawings, in which:

Figure 1 is a block diagram of a conventional web video phone which does not adopt a swapping apparatus;

Figure 2 is a block diagram of an apparatus for swapping between main and auxiliary memories according to a preferred embodiment of the present invention;

Figure 3 is a flowchart illustrating a method for swapping between main and auxiliary memories according to an embodiment of the present invention;

Figure 4 is a block diagram of a web video phone adopting a swapping apparatus according to an embodiment of the present invention; and

Figure 5 is a block diagram showing a main menu displayed in the display shown in Figure 4.

Referring to Figure 2, an apparatus for swapping between a main memory and an auxiliary memory is shown, comprising a DRAM 20, a flash memory 22, a virtual address table 24, and a swapping controller 26. Here, the DRAM 20 is a main memory, and the flash memory 22 is an auxiliary memory.

The flash memory 22 stores a plurality of application programs including first and second application programs 220 and 230. The application program stores a program code and data.

The DRAM 20 includes a basic program code region 202, a program code region 204, and a data region 206.

The basic program code region 202 stores a real time OS (operating system) code and a program code. The program code displays a main menu necessary to interface with a user, and processes a user input.

The program code region 204 is a region into which the program codes (222, 232,...) of the application programs (220, 230,...) in the flash memory 22 are downloaded. It is preferable that the capacity of the program code region 204 is larger than that of each of the program codes (222, 232,...) of the application programs (220, 230,...).

The data region 206 is swapped for the data (224, 234,...) of the application programs (220, 230,...) in the flash memory 22.

The virtual address table 24 stores starting addresses of the application programs (220, 230,...). First and second addresses 242 and 244 are addresses where the first and second amplification programs 220 and 230 start, respectively.

The swapping controller 26 performs the following control in response to an execution command of the first application program 220 by a user interface. First, the swapping controller 26 controls downloading of the program code 1 220 of the flash memory 22 into the program code region 204 of the DRAM 20 with reference to the first address 242 stored in the virtual address table 24. Also, the swapping controller 26 controls swapping of data stored in the data region 206 of the DRAM 20 for the data 1 224 in the flash memory 22.

Details of the control operation of the swapping controller 26 will now be described.

After a system is turned on, it is determined whether the first application program is to be downloaded first. If it is determined that the first application program will be downloaded first, the program code 1 222 is controlled to be downloaded into the DRAM 20, and the data 1 224 of the first application program 220 in the flash memory 22 is controlled not to be downloaded into the DRAM 20.

Upon completion of the first application program, the swapping controller 26 stores data stored in the data region 206 of the DRAM, in the data 1 224. That is, the swapping between data of the DRAM and data of the flash memory occurs when each application program is downloaded into the DRAM two or more times.

Also, the swapping controller 26 controls swapping of data stored in the data region of the DRAM 20 for the data 1 224 stored in the flash memory 22, in response to an execution command with respect to the second application program 230 by a user interface. Also, the program code 2 232 in the flash memory 22 is controlled to be written over the program code 1 220 stored in the program code region 204 of the DRAM 20, and the data 2 234 in the flash memory 22 is controlled to be swapped
for the data stored in the data region of the DRAM 20.

In Figure 2, when each application program has a significantly large capacity, it takes too much time to download the application program into the flash memory 22. This can badly influence the performance of the system, so it is preferable that each application program is small.

Also, preferably, the physical reading/writing of the flash memory 22 is fast.

Figure 3 is a flowchart illustrating a preferred method of swapping between a main memory and an auxiliary memory.

A main menu is displayed in step 302. An LCD 46 shown in Figure 5 displays menu selections such as video phone 462, Internet phone 464, E-mail 466, personal information management system (PIMS) 468, web browser 470, etc.

A determination of whether a key input exists is made, in step 304. User interface means that a desired application to be executed is selected from the menu selections displayed in Figure 5, using a touch screen or an input device such as a key-pad.

The remaining steps of the swapping method will be described, assuming that a user selects the first application program 220 shown in Figure 2.

It is determined whether the program code 1 222 of the selected first application program, i.e., a code of the application program desired to be executed, exists in the program code region 204 of the DRAM 20, in step 306.

If it does not exist, the program code 1 220 stored in the flash memory is downloaded into the program code region 204 of the DRAM referring to the virtual address table 24, in step 308.

Data stored in the data region 206 of the DRAM is stored in a data region of the latest application program

(i.e., which is executed right before the program code 1 is downloaded) in the flash memory, in step 310. If the program code 1 is detected in step 306, step 312 is performed.

It is determined whether the program code 1 loaded in the program code region 204 of the DRAM is a program code first loaded after the system is turned on, in step 312.

If it is determined that the program code 1 is not the first loaded one, the data stored in the data region 206 of the DRAM and the data 1 224 of the first application program are swapped, in step 314.

If it is determined that the program code 1 is the first loaded one, a starting address of the program code 1 222 of the first application program in the DRAM is loaded in a program counter (PC), and the program code 1 222 loaded in the DRAM is executed, in step 316.

It is determined whether an end command with respect to the program code 1 of the first application program is input, in step 318.

If the end command has been input, it is determined whether the currently-executed program code 1 is a code first loaded after the system is turned on, in step 320.

If the program code 1 is the first loaded one, data stored in the data region 206 of the DRAM is stored in the data 1 224 of the first application program in the flash memory, in step 324.

If the program code is not the first loaded one, the data stored in the data region 206 of the DRAM and the

data 1 224 of the first application program are swapped, in step 322.

Figure 4 is a block diagram of a web video phone adopting a swapping apparatus of the type described above.

Referring to Figure 4, the web video phone includes a key input unit 40, a DRAM 42, a flash memory 44, a display 46 and a main processor 48.

The DRAM 42 comprises a basic program code region 422, a program code region 424, and a data region 426. An Internet phone application using the H.323 protocol is stored in the program code region 424.

The flash memory 44 stores a video phone application 442 using the H.324 protocol, a web browser application 444, an e-mail application 446, a PIMS application 448, and an Internet phone application 450 using the H.323 protocol.

The main processor 48 performs the following controls in response to an execution command with respect to an application program input by a user using the key input unit 40 after viewing the menu selections displayed on the display 46. A program code of an application program in the flash memory 44 is controlled to be stored in the program code region 424 of the DRAM 42. Also, the main processor 48 controls the swapping between the data of each application program in the flash memory 44 and data stored in the data region 426 of the DRAM 42.

The operation of the web video phone shown in Figure 4 will now be described, on the assumption that the Internet phone application 450 using the H.323 protocol

was selected through the key input unit 40 among the application programs 442 through 450 stored in the flash memory 44.

When the code of the Internet phone 450 using the H.323 protocol already exists in the program code region 424 of the DRAM 42, the main processor 48 prevents downloading from the flash memory 44 to the DRAM 42. Also, when the Internet phone 450 is a program first loaded after a system is turned on, the main processor 48 prevents data (not shown) of the Internet phone application 450 stored in the flash memory 44 from being downloaded into the DRAM 42. Data stored in the data region 426 of the DRAM upon termination of the Internet phone application 450 is stored in a data region (not shown) of the Internet phone application 450. When the Internet phone application 450 is downloaded two or more times after the system is turned on, the data stored in the data region 426 of the DRAM upon termination and the data stored in the data region (not shown) of the Internet phone application 450 are swapped.

Meanwhile, the embodiment of the present invention can be written in a program capable of being executed in a computer. Also, the embodiment can be realized in a general purpose digital computer which operates the program from media used in the computer. The media can be a magnetic storage media (e.g.: ROM, floppy disks, hard disks, etc.), optical read media (e.g.: CD-ROM, DVD, etc.), or recording media such as a carrier wave (e.g.: transmission through Internet).

In the method of swapping between a main memory and an auxiliary memory, the recording media stores the code

of a program which can perform the following four steps in the computer.

The first step is predetermining whether the program code of an application program to be executed exists in a data region of the main memory. The second step is downloading a program code stored in the auxiliary memory into a program code region of the main memory when it is determined in the first step that the program code does not exist. The third step is swapping data of the application program stored in the auxiliary memory and the data stored in the data region of the main memory. The fourth step is loading into a program counter a start address of the program code downloaded into the main memory in the second step and executing the program code.

The program code further includes a program code which can perform the following fifth and sixth steps.

The fifth step is determining whether the program code downloaded into the main memory is data first loaded after a system is turned on. The sixth step is storing data in the data region of the main memory in the data region of a most recently executed application program, when it is determined in the fifth step that the program code has been first loaded.

The program code further includes a program code which can perform the following fifth and sixth steps.

The fifth step is determining whether the program code downloaded in the second step is data first loaded after a system is turned on, when a termination command of the application program executed in the main memory is input. The sixth step is storing data stored in the data

region of the main memory in the data region of the application program executed in the fourth step, when it is determined in the fifth step that the program code has been first loaded.

The program code further includes a program code which can perform the step of swapping the data stored in the data region of the main memory and the data stored in the data region of the executed application program, when it is determined in the fifth step that the program code has not been first loaded.

Also, the program code further includes a program code which can perform the following fifth and sixth steps.

The fifth step is determining whether an execution command of another application program is input by a user. The sixth step is storing the data stored in the data region of the main memory in the data region of a most recently executed application program and storing data of the another application program in the data region of the main memory, when it is determined in the fifth step that the execution command has been input.

Functional programs, codes and code segments for realizing the present invention can be easily inferred by programmers in a technical field to which the present invention belongs.

The present invention is not limited to the above embodiment, and it is apparent that modifications may be effected by those skilled in the art within the scope of the present invention. That is, in the embodiment of the present invention, a DRAM and a flash memory was taken as

the example of a main memory and an auxiliary memory, respectively. However, a high-speed read/write memory can be used instead of the DRAM, and a low-speed nonvolatile read/write memory can be used instead of the flash memory.

In the method and apparatus described above, only an application program requested by an input device to be executed among many application programs stored in the flash memory is swapped with a program code and data which are executed in a predetermined region of the DRAM. Therefore, all application programs can be driven by a small capacity DRAM, and can be easily, individually managed.

The reader's attention is directed to all papers and documents which are filed concurrently with or previous to this specification in connection with this application and which are open to public inspection with this specification, and the contents of all such papers and documents are incorporated herein by reference.

All of the features disclosed in this specification (including any accompanying claims, abstract and drawings), and/or all of the steps of any method or process so disclosed, may be combined in any combination, except combinations where at least some of such features and/or steps are mutually exclusive.

Each feature disclosed in this specification (including any accompanying claims, abstract and drawings), may be replaced by alternative features serving the same, equivalent or similar purpose, unless expressly stated otherwise. Thus, unless expressly stated otherwise, each feature disclosed is one example only of a generic series of equivalent or similar features.

The invention is not restricted to the details of the foregoing embodiment(s). The invention extends to any novel one, or any novel combination, of the features disclosed in this specification (including any accompanying claims, abstract and drawings), or to any novel one, or any novel combination, of the steps of any method or process so disclosed.

CLAIMS

1. An apparatus for swapping between a main memory and an auxiliary memory, comprising:

an auxiliary memory for storing a plurality of application programs including program codes and data;

a virtual address table for storing a start address of each application program stored in the auxiliary memory;

a main memory having a program code region and a data region for storing a program code downloaded from the auxiliary memory and data swapped for the auxiliary memory; and

a swapping controller for controlling the program codes of the programs to be downloaded into the program code region of the main memory with reference to addresses stored in the virtual address table, and controlling the swapping between the data of the programs and data stored in the data region of the main memory, in response to a request by a user interface to execute the application programs.

2. The apparatus for swapping between a main memory and an auxiliary memory as claimed in claim 1, wherein the main memory is either a DRAM or a high-speed read/write memory and the auxiliary memory is either a flash memory or a low-speed nonvolatile read/write memory.

3. A method for swapping between a main memory and an auxiliary memory, comprising the steps of:

(a) determining whether the program code of an application program to be executed exists in a data region of the main memory;

(b) downloading a program code stored in the auxiliary memory into a program code region of the main memory, when it is determined in step (a) that the program code does not exist;

(c) swapping data of an application program stored in the auxiliary memory for data stored in the data region of the main memory; and

(d) loading to a program counter a start address of the program downloaded into the main memory in step (b) and executing the program code.

4. The method of swapping between a main memory and an auxiliary memory as claimed in claim 3, wherein the program code in step (b) is downloaded referring to a virtual address table which stores the start address of each application program stored in the auxiliary memory.

5. The method of swapping between a main memory and an auxiliary memory as claimed in claim 3, wherein in the step (b), when another program code already exists in the program code region of the main memory, the downloaded program code is written over the existing program code.

6. The method of swapping between a main memory and an auxiliary memory as claimed in claim 3, further comprising the steps of:

(e) determining whether the program code downloaded in the main memory is a code first loaded after a system is turned on; and

(f) storing the data in the data region of the main memory in the data region of a most recently executed application program, when it is determined in step (e) that the program code has been first loaded.

7. The method of swapping between a main memory and an auxiliary memory as claimed in claim 3, further comprising the steps of:

(e) determining whether the program code downloaded in the step (b) is data first loaded after a system is turned on; and

(f) storing data stored in the data region of the main memory in the data region of the application program executed in the step (d), when it is determined in the step (e) that the program code has been first loaded.

8. The method of swapping between a main memory and an auxiliary memory as claimed in claim 7, further comprising the step of swapping the data stored in the data region of the main memory for the data stored in the data region of the executed application program, when it is determined in the step (e) that the program code has not been first loaded.

9. The method of swapping between a main memory and an auxiliary memory as claimed in claim 3, further comprising the steps of:

(e) determining whether an execution command of another application program is input by a user; and

(f) storing the data stored in the data region of the main memory in the data region of a most recently executed application program and storing data of the another application program in the data region of the main memory, when it is determined in the step (e) that the execution command has been input.

10. The method of swapping between a main memory and an auxiliary memory as claimed in claim 9, wherein when it is determined that the program code of the other application program was executed in the program code region of the main memory two or more times, the data of the another application program is swapped for the data in the data region of the main memory.

11. A web video phone wherein the contents of a DRAM and a flash memory are swapped, comprising:

a key input unit for inputting a command of execution of an application program;

a flash memory for storing application programs selected from a group comprising a video phone using the H.324 protocol, an Internet phone using the H.323 protocol, a web browser, an e-mail, and a personal information management system (PIMS);

a DRAM including a program code region and a data region for storing program codes downloaded in the flash memory and data swapped for the flash memory; and

a main processor for controlling the program code of each application program stored in the flash memory to be stored in the program region of the DRAM and controlling the swapping between the data of each application program stored in the flash memory and the data stored in the data region of the DRAM, in response to an execution command of an application program input through the key input unit.

12. An apparatus for swapping between a main memory and an auxiliary memory, substantially as hereinbefore described with reference to Figure 2 of the accompanying drawings.

13. A method for swapping between a main memory and an auxiliary memory, substantially as hereinbefore described with reference to Figures 3 and 5 of the accompanying drawings.

14. A web video phone substantially as hereinbefore described with reference to Figure 4 of the accompanying drawings.

Application No:	GB 9825088.9	**Examiner:**	K. Sylvan
Claims searched:	1-14	**Date of search:**	5 May 1999

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.Q): G4A (AFL)

Int Cl (Ed.6): G06F (9/445)

Other:

Documents considered to be relevant:

Category	Identity of document and relevant passage		Relevant to claims
A	US5371871	Helix Software. See column 2 lines 8-22 and column 3 line 34 to column 4 line 15.	-
A	US5212633	ShareData. See column 9 lines 12-44.	-

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.